

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-161864

(43) 公開日 平成10年(1998) 6月19日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 9/06

識別記号

5 5 0

F I

G 0 6 F 9/06

5 5 0 A

5 5 0 Z

審査請求 未請求 請求項の数 9 O L (全 13 頁)

(21) 出願番号 特願平8-317917

(22) 出願日 平成8年(1996)11月28日

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 田中 利清

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

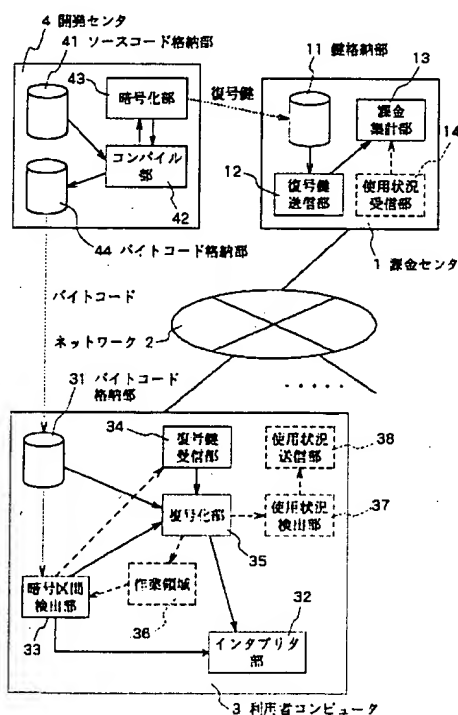
(74) 代理人 弁理士 若林 忠

(54) 【発明の名称】 ソフトウェアの保護方式

(57) 【要約】

【課題】 ソフトウェアの使用機能に着目し、ソフトウェアの使用回数などに応じた使用料の徴収を可能とし、また、不正コピーを防止できる保護方式を提供する。

【解決手段】 ソースコードをコンパイルして得られたバイトコードを利用者コンピュータ3に配布し、利用者コンピュータ3では、バイトコードをインタプリタによって解釈実行するシステムにおいて、配布されるバイトコードの一部を暗号区間として暗号化する。ネットワーク2に接続され暗号区間に対応する復号鍵を保持する課金センタ1を設ける。利用者コンピュータ3には、課金センタ1から復号鍵を取得する復号鍵受信部34と、配布されたバイトコード中での暗号区間の開始と終了とを検出する暗号区間検出部33と、復号鍵を用いて暗号区間内のバイトコードを復号する復号化部35と、復号されたバイトコードを解釈実行するインタプリタ部32と、を設ける。



## 【特許請求の範囲】

【請求項1】 ソフトウェアのソースコードをコンパイラにより変換して得られたバイトコードを利用者コンピュータに配布し、前記利用者コンピュータでは、前記バイトコードをインタプリタによって、順次、解釈実行するソフトウェア実行システムにおける、ソフトウェアの保護方式において、

ソフトウェアごとに前記利用者コンピュータに配布されたバイトコードの少なくとも一部の区間が暗号区間として暗号化されており、

前記利用者コンピュータが接続されたネットワークと、前記ネットワークに接続され前記暗号区間に対応する復号鍵を保持する課金センタとを有し、

前記利用者コンピュータが、前記課金センタから前記復号鍵を取得する復号鍵取得手段と、前記配布されたバイトコード中での前記暗号区間の開始と終了とを検出する暗号区間検出手段と、前記復号鍵を用いて前記暗号区間内のバイトコードを復号する復号手段と、復号手段で復号されたバイトコードを、順次、解釈実行するインタプリタ手段と、を備えていることを特徴とするソフトウェアの保護方式。

【請求項2】 前記利用者コンピュータ内に、当該利用者コンピュータでのソフトウェアの使用を検出して使用記録を保持し、前記使用記録を前記課金センタに送付する使用記録検出手段を有し、前記課金センタが、送付されてきた前記使用記録に基づいて課金処理を実行する、請求項1に記載のソフトウェアの保護方式。

【請求項3】 前記配布されたバイトコードを前記利用者コンピュータが実行する前に前記復号鍵取得手段が前記課金センタから前記復号鍵を取得し、前記課金センタが前記復号鍵の配布履歴に基づいて課金処理を実行する、請求項1に記載のソフトウェアの保護方式。

【請求項4】 前記配布されたバイトコードに複数の暗号区間が設定されて各暗号区間に暗号区間IDが付与され、前記暗号区間IDごとに復号鍵が対応して前記配布されたバイトコードが複数種類の復号鍵に対応し、前記復号鍵取得手段が、前記配布されたバイトコードを前記利用者コンピュータが実行する前に前記課金センタから復号鍵を一括して取得するとともに、前記バイトコードの実行中に、前記暗号区間ごとに前記暗号区間IDに基づいて復号鍵の選択を実行する、請求項1乃至3いずれか1項に記載のソフトウェアの保護方式。

【請求項5】 前記配布されたバイトコードに複数の暗号区間が設定されて各暗号区間に暗号区間IDが付与され、暗号区間IDごとに復号鍵が対応して前記配布されたバイトコードが複数種類の復号鍵に対応し、前記復号鍵取得手段が、前記配布されたバイトコードの実行中に前記暗号区間が検出されるたびに、当該暗号区間の前記暗号区間IDに基づいて前記課金センタから該

当する復号鍵を取得し、

前記課金センタが前記復号鍵の配布履歴に基づいて課金処理を実行する、請求項1に記載のソフトウェアの保護方式。

【請求項6】 前記利用者コンピュータ内に、当該利用者コンピュータでのソフトウェアの使用を検出して使用記録を保持し、前記使用記録を前記課金センタに送付する使用記録検出手段を有し、

前記課金センタが、送付されてきた前記使用記録に基づいて課金処理を実行する、請求項5に記載のソフトウェアの保護方式。

【請求項7】 前記利用者コンピュータに作業領域を有し、前記配布されたバイトコードの実行中に前記暗号区間を検出した場合には、当該暗号区間内の全バイトコードをまとめて復号してから、前記インタプリタ手段が当該バイトコードの解釈実行を行う、請求項1乃至6いずれか1項に記載のソフトウェアの保護方式。

【請求項8】 ソフトウェアのソースコードをコンパイラにより変換して得られたバイトコードを利用者コンピュータに配布し、前記利用者コンピュータでは、前記バイトコードをインタプリタによって、順次、解釈実行するソフトウェア実行システムにおける、ソフトウェアの保護方式において、

前記利用者コンピュータが接続されたネットワークと、前記ネットワークに接続され前記暗号区間に対応する復号鍵を保持する課金センタとを有し、

前記利用者コンピュータ内に、当該利用者コンピュータでのソフトウェアの使用を検出して使用記録を保持し、前記使用記録を前記課金センタに送付する使用記録検出手段が設けられ、

前記課金センタが、送付されてきた前記使用記録に基づいて課金処理を実行することを特徴とするソフトウェアの保護方式。

【請求項9】 ソフトウェアのソースコードをコンパイル手段により変換して得られたバイトコードを利用者コンピュータに配布し、前記利用者コンピュータでは、前記バイトコードをインタプリタによって、順次、解釈実行するソフトウェア実行システムにおける、ソフトウェアの保護方式において、前記コンパイル手段は、暗号区間の開始、暗号区間の終了を示すコードが前記ソースコードに存在する場合には、当該コードを暗号区間の開始、暗号区間の終了を示すバイトコードに変換するものであり、

前記暗号区間内のバイトコードを暗号化する暗号化手段をさらに備え、

ソフトウェアごとに前記利用者コンピュータに配布されるバイトコードの少なくとも一部の区間が前記暗号区間として暗号化されていることを特徴とする、ソフトウェアの保護方式。

【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明はいわゆる中間コードとしてのバイトコードを使用するソフトウェアに対する保護方式に関し、特に、ソフトウェアの使用量に比例した使用料金の徴収を可能とするソフトウェアの保護方式に関する。

## 【0002】

【従来の技術】ソフトウェアの中には、Javaアプレットやある種のアプリケーションなど、ソフトウェアの開発時にはコンパイラによりソースコードをバイトコード（中間コード）に変換し、このバイトコードを利用者に配布し、ソフトウェアの実行時にインタプリタによりこのバイトコードを、順次、解釈実行するものがある。この利用者側のコンピュータには、バイトコードを解釈実行するためのインタプリタを予め用意しておくものとする。この方法では、利用者側のコンピュータに要求されるハードウェアがそれほど大きくなくて済む上、配布されるバイトコード自体も比較的コンパクトなため、ネットワークなどの使用を前提として、例えば、ネットワークに接続された多数のコンピュータにソフトウェアを同時にインストールしたりするのに有効な技術として、注目を集めている。

【0003】ところで、従来、パーソナルコンピュータなどのアプリケーションソフトウェアは、一般に、これらソフトウェアの実行形式プログラムを格納した記録媒体を用いて、買い取り方式により販売されている。しかしながら、買い取り方式では、流通経費を相対的に低減したいという動機づけ（インセンティブ）が働くので、ソフトウェアの機能が必要以上に肥大化し、ほとんど使用しない機能まで含めて利用者は高額な費用を負担せざるを得なくなることが多い。この場合、購入代金は、利用者がそのソフトウェアを実際に使用した頻度によらず、一定である。また、機能が必要以上に肥大化した際には、そのソフトウェアを実行させるために必要なハードウェア量も増大し、動作速度が低下する。さらに、実際にソフトウェアを購入して実行してみなければ、利用者が必要とする機能が満足されているか否かを判断することができない。

【0004】最近では、買い取り方式の変形として、暗号化したソフトウェアをCD-ROMなどの記録媒体またはネットワーク経由で配布し、電話、ファクシミリ、手紙（郵送）または電子メールによる購入手続きの後、復号鍵を通知する方式も採用されている。この場合でも、使用回数にかかわらず同一の金額を支払わなければならない、使用回数または使用時間当たりの価格には大きな幅がある。さらにこの方式では、復号したソフトウェアを不正にコピーして使用することを防ぐことができず、著作権者の権利の保護に万全を期することができない。

## 【0005】

【発明が解決しようとする課題】以上説明したように、アプリケーションソフトウェアの現行の販売方法では、利用者の側には、不要な機能にまで代金を払っているという不満があり、一方、著作権者の側には権利が十分保護されないという不満がある。

【0006】本発明の目的は、ソフトウェアの使用機能に着目し、ソフトウェアの使用量、使用回数などに応じた使用料の徴収を可能とするとともに、ソフトウェアの不正なコピーを防ぐことができるソフトウェアの保護方式を提供することにある。

## 【0007】

【課題を解決するための手段】本発明のソフトウェア保護方式は、ソフトウェアのソースコードをコンパイラにより変換して得られたバイトコードを利用者コンピュータに配布し、利用者コンピュータでは、バイトコードをインタプリタによって、順次、解釈実行するソフトウェア実行システムにおける、ソフトウェアの保護方式において、ソフトウェアごとに利用者コンピュータに配布されたバイトコードの少なくとも一部の区間が暗号区間として暗号化されており、利用者コンピュータが接続されたネットワークと、ネットワークに接続され暗号区間に対応する復号鍵を保持する課金センタとを有し、利用者コンピュータが、課金センタから復号鍵を取得する復号鍵取得手段と、配布されたバイトコード中での暗号区間の開始と終了とを検出する暗号区間検出手段と、復号鍵を用いて暗号区間内のバイトコードを復号する復号手段と、復号手段で復号されたバイトコードを、順次、解釈実行するインタプリタ手段と、を備えている。

【0008】すなわち、本発明では、CD-ROMやフロッピーディスクなどの記録媒体により、あるいは、ネットワークを介して流通するソフトウェアは、バイトコードであって、その一部がソフトウェア開発時に暗号化されている。そして、課金センタを設けて復号鍵を課金センタで集中して管理し、利用者コンピュータ側からの要求にしたがいネットワークを介して復号鍵が配送される。そして利用者コンピュータとして、バイトコードをインタプリタによって1命令ずつ解釈実行するものを使用する。

【0009】本発明によれば、バイトコードの一部を暗号化し、復号鍵はその都度配送するようにしてソフトウェアの実行中に復号処理を行うので、いずれの時点においても、利用者コンピュータ側では、復号されたソフトウェアが磁気ディスクなどのファイルシステム内に格納されることがなく、ソフトウェアを不正にコピーして使用することが防止される。

【0010】課金センタでは、復号鍵の配布履歴に基づき、あるいは、利用者コンピュータ側からネットワークを介して集めた使用記録すなわち使用状況に関するデータに基づいて課金処理を行うことができ、これにより、使用回数などに比例した使用料の徴収が保証される。

【0011】本発明では、暗号化に使用される暗号鍵としてソフトウェア全体で1つの鍵またはソフトウェアの機能の単位で異なる複数の鍵が使用でき、これら暗号鍵に対応して1あるいは複数の復号鍵が使用される。複数の鍵を使用する場合には、ソフトウェア内の暗号区間の開始を指定するバイトコードのオペラントに暗号区間IDが設定され、暗号区間IDに対応させて復号鍵が管理されるようにすればよい。復号鍵の配送に当たっては、適切な認証および暗号化を実施することが好ましい。

【0012】

【発明の実施の形態】次に、本発明の実施の形態について、図面を参照して説明する。図1は本発明の実施の一形態のソフトウェアの保護方式の構成を示すブロック図である。

【0013】このソフトウェアの保護方式では、課金センタ1と利用者コンピュータ（実行マシン）3とが、ネットワーク2を介して接続している。ここでは、利用者コンピュータ3は、アプリケーションソフトウェアとして、実行形式プログラムを直接実行するのではなく、ソースコードをコンパイルして得られたバイトコード（中間コード）が配布され、この配布されたバイトコードをインタプリタによって1命令ずつ解釈して実行するように構成されている。そして、利用者コンピュータ3に配布されるバイトコードは、その一部（暗号区間という）が暗号化されており、暗号区間の復号に必要な復号鍵は、利用者コンピュータ3でのソフトウェアの解釈実行の都度、課金センタ1からネットワーク2を介して利用者コンピュータ3に転送される。また、ソースコードから一部暗号化されたバイトコードを生成し、利用者コンピュータ3に配布するために、開発センタ4が設けられている。開発センタ4から利用者コンピュータ3へのバイトコードの配布は、ネットワークを介してもよいし、CD-ROMなどの記録媒体を介してもよい。図1では利用者コンピュータ3が1つしか描かれていないが、当然のことながら、ネットワーク2には多数の利用者コンピュータ3を接続することができる。

【0014】以下、課金センタ1、利用者コンピュータ3及び課金センタ4の構成について、詳しく説明する。なお、図1において破線で表わされた各構成要素は、本実施の形態における付加的な構成要素である。

【0015】課金センタ1には、利用者コンピュータ3に配布する復号鍵を格納する鍵格納部11と、利用者コンピュータ3側からの要求に応じてネットワーク2を介し復号鍵をその利用者コンピュータ3に送信する復号鍵送信部12と、課金の集計処理を行う課金集計部13とが設けられている。さらに、ネットワーク2を介して利用者コンピュータ3がその利用者コンピュータ3でのソフトウェアの使用状況を送信する場合には、この使用状況のデータを受信するために、使用状況受信部14を課金センタ1内に設ければよい。課金集計部13は、復号鍵

送信部12での復号鍵の送信の履歴に応じ、使用状況受信部14が設けられている場合には受信した使用状況のデータに応じ、利用者ごとの課金を集計する。

【0016】利用者コンピュータ3は、開発センタ4で生成したアプリケーションソフトウェア（課金対象のソフトウェア）のバイトコードを格納するバイトコード格納部31と、バイトコードを解釈実行するインタプリタ部32と、バイトコード格納部31に格納されたバイトコード中の暗号区間を検出する暗号区間検出部33と、課金センタ1に対して復号鍵の送信を要求しネットワーク2を介して送信されてきた復号鍵を受信する復号鍵受信部34と、復号鍵受信部34で受信した復号鍵に基づき、暗号区間検出部33で検出された暗号区間のバイトコードを復号する復号化部35とを備えている。この利用者コンピュータ3には、復号化部35で復号された、暗号区間のバイトコードを一時的に格納する作業領域を設けるようにしてもよい。この作業領域は、例えば利用者コンピュータ3の主記憶メモリ上の領域として確保することができる。また、課金センタ1側にソフトウェアの使用状況を通知するために、復号化部35での復号処理の実行を検出してソフトウェアの使用状況を検出して記憶する使用状況検出部37と、検出された使用状況に関するデータを課金センタ1側にネットワークを介して送信するための使用状況送信部38とを、利用者コンピュータ3内に設けるようにしてもよい。

【0017】利用者コンピュータ3において、バイトコード格納部31は、例えば、ハードディスク装置などの記憶装置によって構成される。また、インタプリタ部32は、暗号区間でない部分のバイトコードについては、バイトコード格納部31から読み出されたバイトコードをそのまま解釈実行し、暗号区間のバイトコードについては、復号されて復号化部35から出力されるバイトコード（作業領域36が設けられている場合は復号され作業領域36に格納されたバイトコード）を解釈実行する。インタプリタ部32は、具体的には、バイトコードを1命令ずつ解釈して実行するためのインタプリタと呼ばれるソフトウェアと、インタプリタを実行するCPU（中央処理装置）などによって、構成される。また、復号鍵受信部34が復号鍵の送信を要求するのは、例えば、暗号区間を含むバイトコードで表わされるソフトウェアを起動するとき、あるいは、実行中のソフトウェアのバイトコードで暗号区間を検出したとき、などである。

【0018】利用者コンピュータ3は、パーソナルコンピュータなどの形態をとるものであって、ここでは具体的には図示していないが、キーボードやマウスなどの入力手段、液晶ディスプレイやCRTなどの出力手段、インタプリタ部32の構成要素ともなるCPU、さらには、メモリ、各種の記憶装置などを備えている。

【0019】開発センタ4は、ソースコードを格納する

ソースコード格納部41と、ソースコード格納部41に格納されたソースコードをコンパイルしてバイトコードとするコンパイル部42と、コンパイル部42の指示によってバイトコードの一部の暗号化処理を行う暗号化部43と、生成したバイトコードを格納するバイトコード格納部44とを備えている。暗号化部43での暗号化に対応する復号鍵が、課金センタ1の鍵格納部11に格納される。バイトコードでの暗号区間の指定は、ソースコードに、暗号区間の開始や終了を示すソースコードを記述することでを行い、コンパイル部42は、このようなソースコードを検出した場合、暗号区間の開始や終了を表わすバイトコードを生成するとともに、この区間内のバイトコードを暗号化部43に渡して暗号化させ、暗号化後のバイトコードをバイトコード格納部44内に格納する。

【0020】以上説明したシステムにおいて、復号鍵受信部34は復号鍵取得手段であり、暗号区間検出部33は暗号区間検出手段であり、復号化部35は復号手段であり、インタプリタ部32はインタプリタ手段であり、使用状況検出部37及び使用状況送信部38は使用記録検出手段である。さらに、コンパイル部42がコンパイル手段であり、暗号化部43が暗号化手段である。

【0021】次に、上述したソフトウェアの保護方式の動作について、具体的な実施例を用いて説明する。

【0022】《実施例1》実施例1は、基本構成、すなわち、使用状況受信部14や作業領域36、使用状況検出部37、使用状況送信部38を含まない構成についてのものである。実施例1での処理の流れが図2に示されている。

【0023】課金対象のソフトウェアが、予めバイトコードとして利用者コンピュータ3のバイトコード格納部31に格納されている。このバイトコードでは、予め一部が暗号化されており、上述したように、暗号区間の開始及び終了を示すバイトコードが埋め込まれている。暗号区間は1箇所または複数箇所存在するが、ここでは、暗号区間が複数箇所存在する場合でも、復号のための復号鍵は1種類であるとする。

【0024】利用者コンピュータ3では、課金対象のアプリケーションソフトウェアが起動されると（ステップ101）、復号鍵受信部34が、このソフトウェアの実行前に、ネットワーク2を介し、課金センタ1からこのソフトウェアの復号鍵を取得する（ステップ102）。復号鍵の配送には、適切な認証及び暗号化を実施するものとする。暗号区間検出部33は、暗号区間が開始したかどうか、すなわち暗号区間開始を表わすバイトコード（暗号区間開始バイトコード）があるかを常時監視している（ステップ103）。暗号区間開始でない場合には、インタプリタ部32が、1命令ずつバイトコードをバイトコード格納部31から読み出し、解釈実行し（ステップ104）、そののちステップ103に戻る。ステ

ップ104を含むループは、暗号区間開始バイトコードを検出するまで繰り返される。

【0025】ステップ103で暗号区間開始バイトコードを検出した場合、すなわち暗号区間が開始した場合には、暗号区間検出部33は、暗号区間終了であるかどうか、すなわち暗号区間終了を示すバイトコード（暗号区間終了バイトコード）を常時監視する（ステップ105）。暗号区間終了であればステップ103に戻り、暗号区間終了でなければ、復号化部35が、課金センタから復号鍵受信部34が取得した復号鍵を用いて、1命令ずつ読み出されたバイトコードを復号し（ステップ106）、このように復号されたバイトコードをインタプリタ部32が解釈実行し（ステップ107）、ステップ105に戻る。このステップ106及びステップ107を含むループは、暗号区間終了バイトコードを検出するまで、すなわち暗号区間終了を検出するまで繰り返される。

【0026】結局、暗号区間終了バイトコードを検出した場合には、暗号区間開始バイトコードを再度検出するまで、復号を行うことなしに、インタプリタ部32が1命令ずつバイトコードを読み出して解釈実行する。

【0027】以上、利用者コンピュータ3での処理を説明したが、このとき、課金センタ1では、復号鍵送信部12による復号鍵の送信履歴に基づき、課金集計部13が利用者ごとの課金処理を実行する。

【0028】《実施例2》実施例1ではアプリケーションプログラムごとに1つの復号鍵が用いられていたが、この実施例2は、アプリケーションプログラムのバイトコード中に複数の暗号区間が設定されるとともに、典型的には暗号区間ごとに異なる復号鍵を用いることにより、全体として複数の復号鍵が用いられていることで、実施例1と異なっている。暗号区間ごとにそこで使用すべき復号鍵を特定するために、暗号区間開始バイトコードには暗号区間を識別する暗号区間IDが埋め込まれており、暗号区間IDとその暗号区間を復号するための復号鍵が対で管理されている。その他の点では、この実施例2は実施例1と同様の構成である。実施例2での処理の流れが図3に示されている。

【0029】利用者コンピュータ3では、課金対象のアプリケーションソフトウェアが起動されると（ステップ111）、復号鍵受信部34が、このソフトウェアの実行前に、ネットワーク2で接続された課金センタ1から、暗号区間IDと対応付けられた復号鍵を複数個一括して取得する（ステップ112）。暗号区間検出部33は、暗号区間が開始したかどうかを常時監視しており（ステップ113）、暗号区間開始でない場合は、実施例1と同様に、インタプリタ部32は、1命令ずつバイトコードを解釈実行し（ステップ114）、ステップ113に戻る。

【0030】ステップ113で暗号区間が開始した場合

には、復号鍵受信部34は、暗号区間開始バイトコードで指定された暗号区間IDに対応した復号鍵を選択する(ステップ115)。そして、暗号区間終了であるかどうか判断され(ステップ116)、暗号区間終了であればステップ113に戻り、暗号区間終了でなければ、復号化部35が、ステップ115で選択された復号鍵によって、1命令ずつ読み出されたバイトコードを復号し(ステップ117)、復号されたバイトコードをインタプリタ部32が解釈実行し(ステップ118)、ステップ116に戻る。このステップ117及びステップ118を含むループは、暗号区間終了を検出するまで繰り返される。

【0031】この実施例2でも、課金センタ1では、復号鍵送信部12による復号鍵の送信履歴に基づき、課金集計部13が利用者ごとの課金処理を実行する。

【0032】《実施例3》この実施例3は、実施例2と同様のものであるが、アプリケーションソフトウェアの実行開始後に暗号区間ごとの復号鍵をその都度取得する点で、実施例2と異なっている。実施例3での処理の流れが図4に示されている。この実施例3でも、アプリケーションソフトウェアには複数の暗号区間が設定され、それぞれ異なる暗号区間IDで特定される複数の復号鍵が使用され、暗号区間IDが暗号区間開始バイトコードに埋め込まれている。

【0033】利用者コンピュータ3では、課金対象のアプリケーションソフトウェアが起動されると(ステップ121)、そのまま、このアプリケーションプログラムの実行が開始される。暗号区間検出部33は、暗号区間が開始したかどうかを常時監視しており(ステップ122)、暗号区間開始でない場合は、実施例1と同様に、インタプリタ部32が1命令ずつバイトコードを解釈実行し(ステップ123)、ステップ122に戻る。

【0034】ステップ122で暗号区間が開始が検出された場合には、復号鍵受信部34が、暗号区間開始バイトコードで指定された暗号区間IDに対応した復号鍵の送信を課金センタ1側に要求し、その鍵を取得する(ステップ124)。そして、暗号区間終了であるかどうか判断され(ステップ125)、暗号区間終了であればステップ122に戻り、暗号区間終了でなければ、復号化部35が、ステップ124で取得された復号鍵によって、1命令ずつ読み出されたバイトコードを復号し(ステップ126)、復号されたバイトコードをインタプリタ部32が解釈実行し(ステップ127)、ステップ125に戻る。このステップ126及びステップ127を含むループは、暗号区間終了を検出するまで繰り返される。

【0035】この実施例3では、暗号区間IDで特定される暗号区間ごとの復号鍵の送信履歴に基づき、課金集計部13での利用者ごとの課金処理が実行される。したがって、アプリケーションプログラムに複数の機能が含まれているとして、各機能ごとの使用頻度に応じた課金

が可能となる。具体的には、利用者は、その使用しなかった機能に対する使用量を払う必要がなくなる。

【0036】《実施例4》実施例1では、復号鍵を用いた復号の際、バイトコードの1命令ごとに、復号と解釈実行が行われている(図1のステップ117、118のループ)。これに対し、この実施例4では、暗号区間を検出したらその暗号区間内のバイトコードを一括して復号して作業領域36に格納し、その後、インタプリタ部32が復号されたバイトコードを1命令ずつ実行する。その他の点では、実施例4は実施例1と同様の構成であり、復号鍵の種類は、暗号区間が複数ある場合でも、1つである。図5は実施例4での処理を示す流れ図である。

【0037】利用者コンピュータ3では、アプリケーションソフトウェアが起動されると(ステップ131)、復号鍵受信部34が、このソフトウェアの実行前に、ネットワーク2で接続された課金センタ1から、復号鍵を取得する(ステップ132)。暗号区間検出部33は、暗号区間が開始したかどうかを常時監視しており(ステップ133)、暗号区間開始でない場合は、実施例1と同様に、インタプリタ部32が1命令ずつバイトコードを解釈実行して(ステップ134)、ステップ133に戻る。

【0038】ステップ133で暗号区間の開始を検出した場合には、暗号区間の終了となったかどうかを常時監視するようにし(ステップ135)、暗号区間終了でない場合には、復号鍵受信部34が受信した復号鍵に基づいて、復号化部32が1命令ずつバイトコードを復号し(ステップ136)、復号したバイトコードを作業領域36内に格納し(ステップ137)、ステップ135に戻る。すなわち、暗号区間終了となるまで、ステップ135～137のループが繰り返して実行され、その暗号区間のバイトコードが一括して復号されて作業領域36内に格納されることになる。そして、ステップ135で暗号区間終了を検出したら、今度はインタプリタ部32が作業領域36から1命令ずつバイトコードを取り出し(ステップ138)、作業領域36から取り出したバイトコードが暗号区間終了バイトコードであるかどうかを判断する(ステップ139)。暗号区間終了バイトコードでなければ、インタプリタ部32は、作業領域36から取り出した1命令分のバイトコードを解釈実行し(ステップ140)、ステップ138に戻る。すなわち、インタプリタ部32は、作業領域36内の復号されたバイトコードを連続して解釈実行する。ステップ139で暗号区間終了を検出した場合には、ステップ133に戻る。

【0039】この実施例4では、暗号区間ごとにバイトコードを一括して復号して作業領域36に格納するので、バイトコードの復号に伴うスループットを向上させることができる。

【0040】《実施例5》上述の各実施例では、復号鍵

の配送に着目して課金処理を行っているが、この実施例5では、課金センタ1内に使用状況受信部14を設け、利用者コンピュータ3内に使用状況検出部37と使用状況送信部38を設け、利用者コンピュータ3から課金センタ1にネットワークを介して送信される使用状況データに基づいて、課金処理が行われるようにしている。ここでは、実施例1と同様に、復号鍵の種類は1種類であって、ソフトウェアの実行前に復号鍵受信部34が復号鍵を取得するものとする。実施例5での処理の流れが図6に示されている。

【0041】利用者コンピュータ3では、課金対象のアプリケーションソフトウェアが起動されると（ステップ151）、復号鍵受信部34が、このソフトウェアの実行前に、ネットワーク2で接続された課金センタ1から、復号鍵を取得する（ステップ152）。暗号区間検出部33は、暗号区間が開始したかどうかを常時監視しており（ステップ153）、暗号区間開始でない場合は、実施例1と同様に、インタプリタ部32が、1命令ずつバイトコードを解釈実行し（ステップ154）、ステップ153に戻る。

【0042】ステップ153で暗号区間が開始した場合には、使用状況検出部37がソフトウェアの使用記録を取得し、記憶する（ステップ155）。そして、暗号区間終了であるかどうか判断され（ステップ156）、暗号区間終了であればステップ153に戻り、暗号区間終了でなければ、復号化部35が、復号鍵受信部34によって取得した復号鍵によって、1命令ずつ読み出されたバイトコードを復号し（ステップ157）、復号されたバイトコードをインタプリタ部32が解釈実行し（ステップ158）、ステップ156に戻る。このステップ157及びステップ158を含むループは、暗号区間終了を検出するまで繰り返される。

【0043】一方、使用状況検出部37で取得され記憶された使用状況のデータは、使用状況受信部38により、上記処理とは非同期に例えば所定の期間ごとに、ネットワーク2を介して課金センタ1の使用状況受信部14に送信される（ステップ159）。課金センタ1では、受信した使用状況データによって、利用者ごとの課金処理が実行される。

【0044】この実施例5では、復号鍵の配送ごとに課金処理を行う必要がないので、課金センタ1での処理が軽減される。

【0045】《実施例6》この実施例6は、開発センタ4での処理に係るものである。図7は実施例6での処理を示す流れ図である。

【0046】ソースコード格納部41にはソースコードが格納されており、ソースコード中では、暗号区間の開始と終了とがそれぞれ暗号区間開始ソースコードと暗号区間終了ソースコードで示されているものとする。ソースコードのコンパイルが開始すると（ステップ16

1）、コンパイル部42は、ソースコードをバイトコードに変換する（ステップ162）。その際、暗号区間開始ソースコード及び暗号区間終了ソースコードは、それぞれ、暗号区間開始バイトコード及び暗号区間終了バイトコードに変換される。そして、暗号化部43により、予め指定された暗号鍵を用いて、暗号区間開始バイトコードと暗号区間終了バイトコードとの間の区間、すなわち暗号区間のバイトコードを暗号化する（ステップ163）。このように一部が暗号化されたバイトコードは、バイトコード格納部44内に格納され、ネットワークやCD-ROMなどを用いて、利用者コンピュータ3に配布される。また、上記の暗号鍵に対応する復号鍵は、課金センタ1の鍵格納部11内に格納される。

【0047】

【発明の効果】以上説明したように本発明によれば、利用者コンピュータにおいて、いずれの時点においても、暗号化されていない部分を含まないソフトウェアがハードディスク等の記憶媒体上に存在しない。このため、ソフトウェアの不正な反復使用を確実に阻止することができ、著作権者はソフトウェアの使用のつど確実に使用料を徴収することが可能になり、ユーザにとっても必要な機能のみを安価な使用料で使用することが可能になるという効果がある。また、ソフトウェアの暗号化に用いる暗号鍵の個数と暗号化の範囲を選択することにより、起動するソフトウェアの単位で課金することも、起動するソフトウェア内の機能単位で課金することも可能となる。

【0048】暗号区間の復号による性能低下は、暗号区間全体を一度に復号して作業記憶に格納することにより、軽減できる。さらに、暗号区間に入った時点で使用記録を取得し、ソフトウェア使用終了後あるいは一定期間間隔で使用記録を課金センタに送付することにより、きめ細かい使用記録に基づく課金が、課金センタとの通信のオーバーヘッドを増やすことなく実現可能となる。

【0049】さらにまた、コンパイラに暗号区間開始／終了のソースコードを解釈し暗号化する機能を付加することにより、高い保護特性を持つソフトウェアの自動生成が可能となる。

【図面の簡単な説明】

【図1】本発明の実施の一形態のソフトウェアの保護方式の構成を示すブロック図である。

【図2】実施例1での利用者コンピュータにおける処理を説明する流れ図である。

【図3】実施例2での利用者コンピュータにおける処理を説明する流れ図である。

【図4】実施例3での利用者コンピュータにおける処理を説明する流れ図である。

【図5】実施例4での利用者コンピュータにおける処理を説明する流れ図である。

【図6】実施例5での利用者コンピュータにおける処理

を説明する流れ図である。

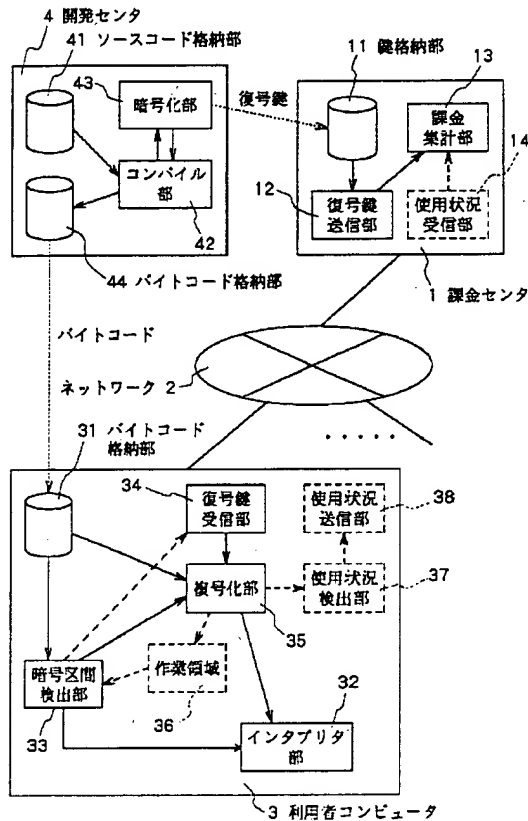
【図7】実施例6での開発センタにおける処理を説明する流れ図である。

【符号の説明】

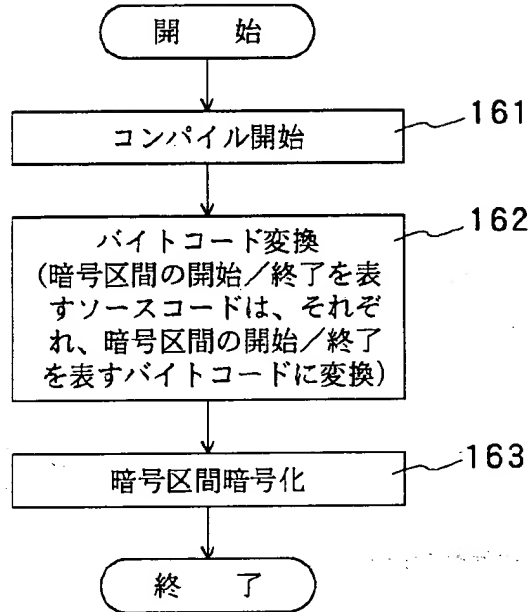
- 1 課金センタ
- 2 ネットワーク
- 3 利用者コンピュータ
- 4 開発センタ
- 11 鍵格納部
- 12 復号鍵送信部
- 13 課金集計部
- 14 使用状況受信部
- 31、44 バイトコード格納部
- 32 インタプリタ部

- 33 暗号区間検出部
- 34 復号鍵受信部
- 35 復号化部
- 36 作業領域
- 37 使用状況検出部
- 38 使用状況送信部
- 41 ソースコード格納部
- 42 コンパイル部
- 43 暗号化部
- 101～107, 111～118, 121～127  
ステップ
- 131～140, 151～159, 161～163  
ステップ

【図1】

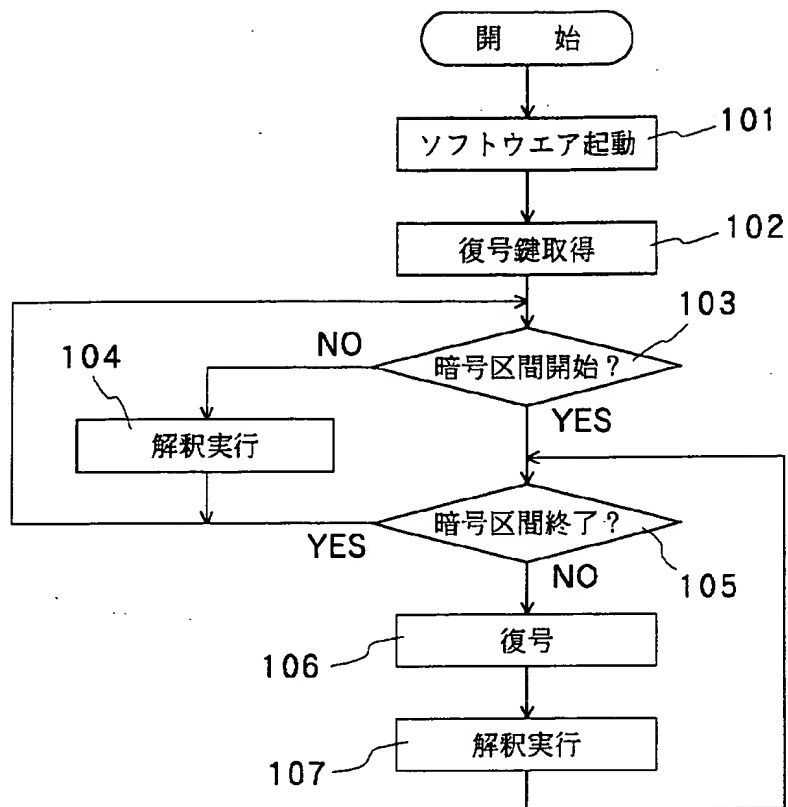


【図7】

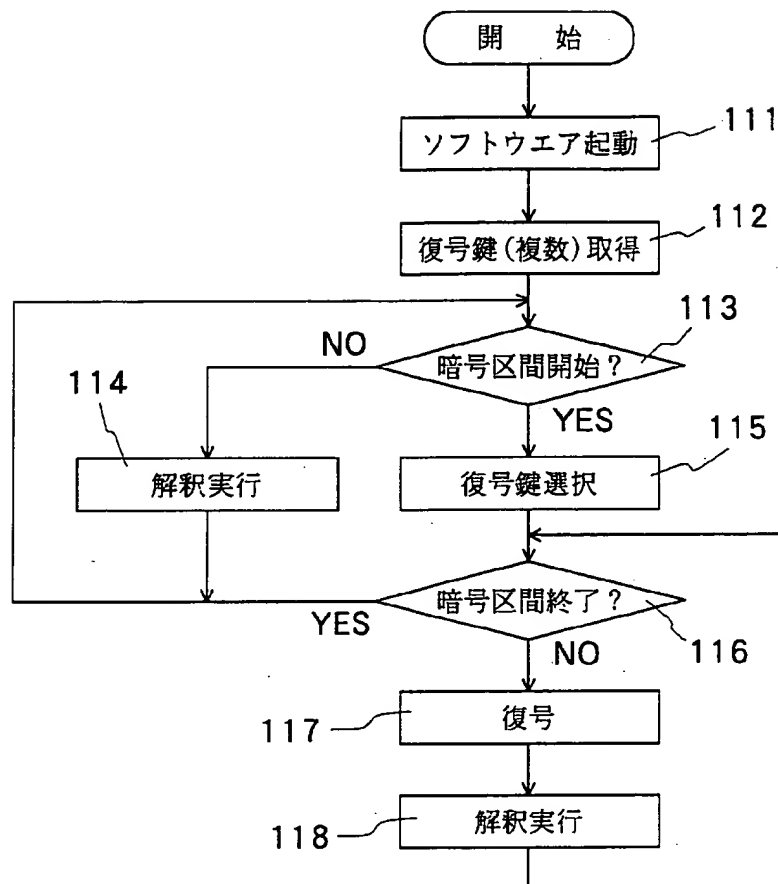




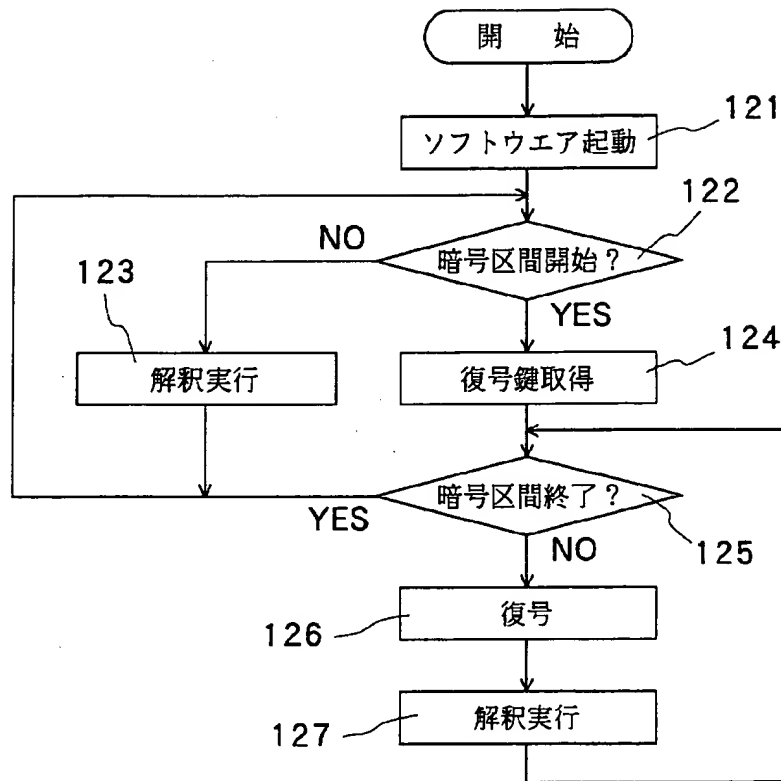
【図2】



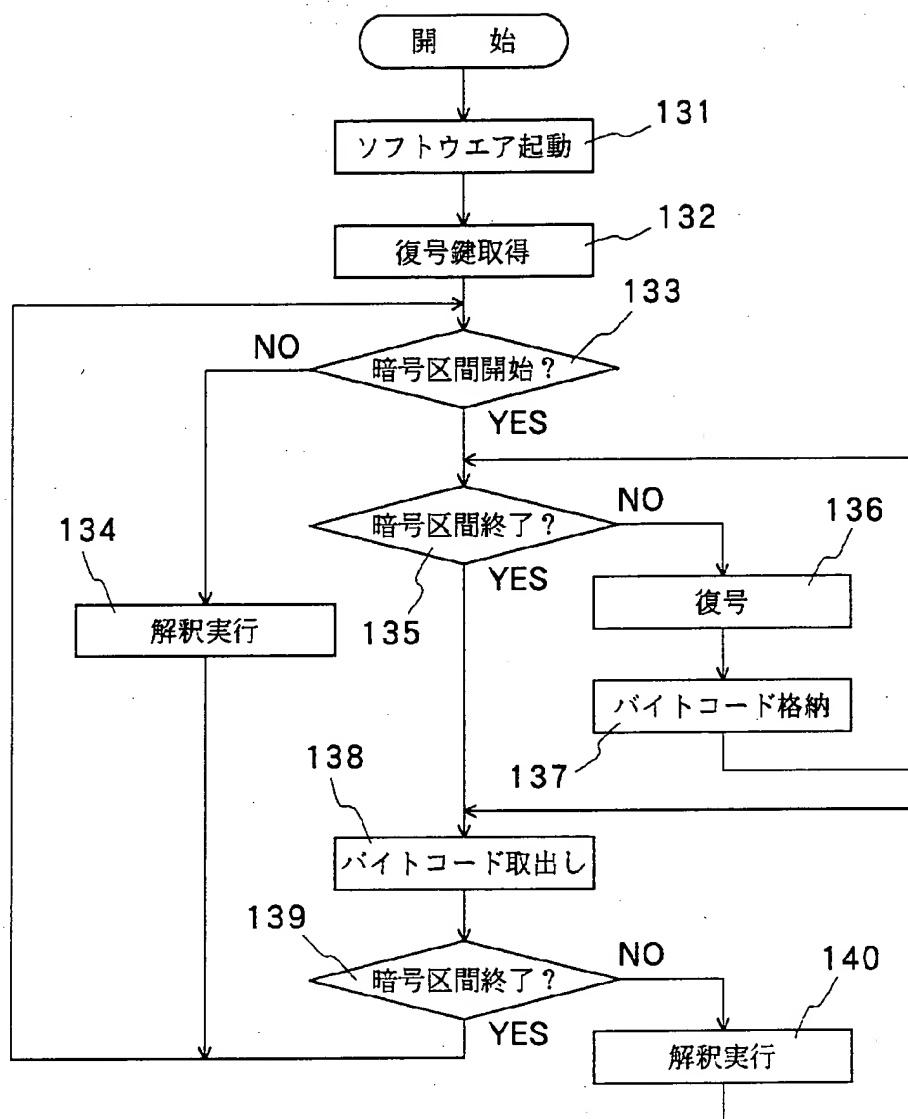
【図3】



【図4】



【図5】



【図6】

